CSE509 : Computer System Security

Covert Channels and Side-Channel Attacks

Covert Channels

- Confidential information may be leaked via channels that may be missed easily
  - Implicit flows in a program
  - Timing channels (network, cache, ...)
  - Steganographic techniques
- Examples
  - transmit info by file name or metadata (e.g., timestamp)
    - Information retrieved by checking file presence or stat
      - No need to read the file (or have read permissions on the file)
  - "Port-knocking"
    - Transmit info by probing network ports in a certain sequence
  - tcp acks or retransmissions, packet fragmentation, …

Side-channel attacks

- Critical info may be leaked inadvertently
  - Error messages, e.g., invalid username vs password
  - Timing information
    - How long it took to verify a password, or encrypt something
    - Cache eviction attacks
    - Meltdown and Spectre attacks
  - Power-monitoring attacks
    - Use thermal imaging of a chip to monitor which circuits are being used and/ or how much power is being used
    - Or simply monitor the power supply
  - Differential fault analysis
    - Force a particular fault (e.g., make a data line to be a "1" always) and examine how the program changes its behavior.
    - Rowhammer attacks on DRAM
  - Last two attacks motivate tamper-resistance in the context of building secure devices
    - Military equipment used in the field
    - Other devices that carry secrets and may be lost

Emanations

- Electromagnetic emanations
  - In old days, CRTs produced a lot of emanations that can be used to figure out what someone is doing from a distance
- Keyboard emanations
  - Researchers have shown it is possible to steal passwords using a microphone in a nearby office!
- Power-line emanations
  - Correlates fluctuations in power use (or EM waves on the powerline) with computations being performed
- Snooping using telescopes
  - Not just on-screen images, but reflections on a cup etc.

Remanence

- malloc after free, or reuse of stack variables
  - Exposes secrets that may be private to one program component to another.
- Allocation of physical page for one process after it is used by another process
  - Exposes secrets across processes
  - Can be avoided by immediately erasing confidential data
    - Beware: the compiler may eliminate this during optimization
    - Cache contents are flushed across process switch, so not a problem
- Retained memory contents after power off
- Residual effects on hard drives
  - may be data is just unlinked, not even overwritten
  - even after overwrite, it is often possible to recover old data

Questions?